blog/eigenvalues_in_opencv

Eigenvalues in OpenCV

Finding Eigenvalues and Eigenvectors of a matrix is the most important task in the Eigenfaces algorithm (I mean 50% of the word are made up by "Eigen"...). So if you are working with OpenCV, here is how to do it. Note, that this post solves the eigenvalue problem for symmetric matrices only. If you want to solve the Eigenvalue problem for general matrices, you could use the solver I have ported from the JAMA project to C++ with OpenCV: decomposition.hpp.

OpenCV C API

OpenCV 2.1

... using double precision:

```
double a[] = {
     1.96 , -6.49 , -0.47 , -7.20 , -0.65,
    -6.49 ,  3.80  ,-6.39  , 1.50 , -6.34,
    -0.47 , -6.39 ,  4.17 , -1.51 ,  2.67,
    -7.20 ,  1.50 , -1.51 ,  5.70 ,  1.80,
    -0.65 , -6.34 ,  2.67 ,  1.80 , -7.10};

CvMat mat = cvMat(5,5,CV_64FC1, a);
CvMat* evec  = cvCreateMat(5,5,CV_64FC1);
CvMat* eval  = cvCreateMat(1,5,CV_64FC1);

cvZero(evec);
cvZero(eval);

cvEigenVV(&mat, evec, eval, DBL_EPSILON, 0, 0);
// print matrix
for(int i = 0; i < eval->rows; i++ )
{
    for(int j = 0; j < eval->cols; j++ )
    {
        CvScalar scal = cvGet2D( eval, i, j );
        printf( "%f\t", scal.val[0]);
    }
    printf( "\n" );
}

cvReleaseMat(&evec);
cvReleaseMat(&eval);
```

or using floating point:

```
float a[] = {
     1.96 , -6.49 , -0.47 , -7.20 , -0.65,
    -6.49 ,  3.80  ,-6.39  , 1.50 , -6.34,
    -0.47 , -6.39 ,  4.17 , -1.51 ,  2.67,
    -7.20 ,  1.50 , -1.51 ,  5.70 ,  1.80,
    -0.65 , -6.34 ,  2.67 ,  1.80 , -7.10};

CvMat mat = cvMat(5,5,CV_32FC1, a);
CvMat* evec  = cvCreateMat(5,5,CV_32FC1);
CvMat* eval  = cvCreateMat(1,5,CV_32FC1);

cvZero(evec);
cvZero(eval);

// print matrix
for(int i = 0; i < eval->rows; i++ )
{
```

```
    for(int j = 0; j < eval->cols; j++ )
    {
        CvScalar scal = cvGet2D( eval, i, j );
        printf( "%f\t", scal.val[0]);
    }
    printf( "\n" );
}
```

OpenCV 2.0

If you try the above in the OpenCV 2.0 C API you will only get the greatest Eigenvalue. Here is how to calculate Eigenvalues and Eigenvectors in OpenCV 2.0.

```
float a[] = {
     1.96 , -6.49 , -0.47 , -7.20 , -0.65,
    -6.49 ,  3.80  ,-6.39  , 1.50 , -6.34,
    -0.47 , -6.39 ,  4.17 , -1.51 ,  2.67,
    -7.20 ,  1.50 , -1.51 ,  5.70 ,  1.80,
    -0.65 , -6.34 ,  2.67 ,  1.80 , -7.10};

CvMat mat = cvMat(5,5,CV_32FC1, a);

CvMat* evec  = cvCreateMat(5,5,CV_32FC1);
CvMat* eval  = cvCreateMat(5,1,CV_32FC1);

cvZero(evec);
cvZero(eval);

cvEigenVV(&mat, evec, eval, DBL_EPSILON, -1, -1);
```

OpenCV C++ API

Using the OpenCV C++ API is self explaining:

```
double b[5][5] = {
    { 1.96 , -6.49, -0.47, -7.20, -0.65},
    { -6.49,  3.80, -6.39,  1.50, -6.34},
    { -0.47, -6.39,  4.17, -1.51,  2.67},
    { -7.20,  1.50, -1.51,  5.70,  1.80},
    { -0.65, -6.34,  2.67,  1.80, -7.10}
    };

cv::Mat E, V;
cv::Mat M(5,5,CV_64FC1,b);
cv::eigen(M,E,V);

// eigenvalues sorted desc
for(int i=0; i < 5; i++)
        std::cout << E.at<double>(0,i) << " \t";
```

Python

With Python Bindings:

```
m = []

m.append([1.96 , -6.49 , -0.47 , -7.20 , -0.65])
m.append([-6.49 ,  3.80  ,-6.39  , 1.50 , -6.34])
m.append([-0.47 , -6.39 ,  4.17 , -1.51 ,  2.67])
m.append([-7.20 ,  1.50 , -1.51 ,  5.70 ,  1.80])
m.append([-0.65 , -6.34 ,  2.67 ,  1.80 , -7.10]);

mat = cv.CreateMat(5,5,cv.CV_32FC1)
evectors = cv.CreateMat(5,5,cv.CV_32FC1)
evalues = cv.CreateMat(5,1,cv.CV_32FC1)

for i in range(5):
```

```
    for j in range(5):
        mat[i,j] = m[i][j]

cv.EigenVV(mat, evectors, evalues, 1e-20)

for i in range(5):
        print [evectors[i,j] for j in range(5)]

print [evalues[i,0] for i in range(5)]
```

Results

Octave gives us an ascending ordering of eigenvalues:

```
octave:2> [v,d] = eig(a)
v = ...
d =

  -11.06558    0.00000    0.00000    0.00000    0.00000
    0.00000   -6.22875    0.00000    0.00000    0.00000
    0.00000    0.00000    0.86403    0.00000    0.00000
    0.00000    0.00000    0.00000    8.86546    0.00000
    0.00000    0.00000    0.00000    0.00000   16.09484
```

Result (plus the GNU Scientific Library (GSL) Solver for Symmetric Matrices). Does that look fine?

```
OpenCV C      16.094837       8.865457       0.864028       -6.228747       -11.065575
OpenCV C++    16.0948         8.86546        0.864028       -6.22875        -11.0656
GSL           16.0948         8.86546        0.864028       -6.22875        -11.0656
```

2010-07-15T18:35:00 · Philipp Wagner opencv